34. (New) The bus system of claim 20, further comprising:

at least one connection to at least one of a DFP, an FPGA, and a DPGA.--

Remarks

This Preliminary Amendment cancels, without prejudice, original claims 1-18, and adds new claims 19-34. The new claims conform the claims to U.S. Patent and Trademark Office rules and do not add new matter.

The substitute specification and the amendments to the abstract are to conform the specification and abstract to U.S. Patent and Trademark Office rules. The substitute specification and the amendments to the abstract do not add new matter to the application.

Applicant asserts that the present invention is new, non-obvious, and useful. Prompt consideration and allowance of the present application are earnestly solicited.

Respectfully submitted,

Dated: 24 March 1998 By: _____

Michelle M. Carniaux
Reg. No. 36,098

KENYON & KENYON
One Broadway
New York, NY 10004
(212) 425-7200

I/O and Memory Bus System for DFPs and Units with Two- or Multi-Dimensional Programmable Cell Architectures

[1. Background of the] Field Of The Invention

The present invention

[1.1 Background art

1.1.1 ... in DFP-based systems

In DFPs according] relates to I/O and memory bus systems.

Background Information

As descibed in German Patent [DE 44 16 881 A1, the lines of each edge cell, i.e., a cell at the edge of a cell array, often in direct contact with the terminals of the unit, are guided out via the terminals of the unit] No. [The lines do not have any specific function, and instead they assume the function written in] DE 44 16 881 A1, a data flow processor ("DFP") is described in which the lines of each edge cell (i.e., a cell at the edge of a cell array, often in direct contact with the terminals of the unit) are guided out via the terminals of the unit. The lines do not have any specific function, and instead they assume the function implemented in the edge cells. If several DFPs are interconnected, all terminals are connected to form a matrix.

[1.1.2 ... in systems with two- or multi-dimensional programmable cell architectures (FPGAs, DPGAs)]

In systems with two- or multi-dimensional programmable cell architectures (FPGAs, DPGAs), a certain subset of internal bus systems and lines of the edge [CELLS] cells are connected to the outside via the unit terminals. The

lines do not have any specific function, and instead they assume the function written in the edge cells. If several FPGAs/DPGAs are interconnected, the terminals assume the function implemented in the hardware or software.

[1.2Problems

1.2.1 ... in DFP-based systems]

The wiring expense for the periphery or for interconnecting DFPs is very high, because the programmer must [also] ensure [at the same time] that the respective functions are integrated into the cells of the DFP(s) at the same time. For connecting a memory, a memory management unit must be integrated into the unit. [For connection]Connection of peripherals[, these must] must also be supported[, just as cascading of DFPs must be similarly taken into account.] [The expense is relatively high, and at the same time, area on the unit is lost for the respective implementations]Additionally, the cascading of DFPs must be similarly taken into account.

[1.2.2 ... in systems with two- or multi-dimensional programmable cell architectures (FPGAs, DPGAs)]

[The above]Not only is the expense relatively high, but also [applies to FPGAs and DPGAs, in particular when they are used]area on the unit is lost for [implementation of algorithms and when they work as arithmetic (co)processors]the respective implementations.

[1.3Improvement through the invention; object

The expense of wiring, in particular the number of unit terminals, is greatly reduced]This also applies to FPGAs and DPGAs, particularly when they are used for

implementation of algorithms and when they work as arithmetic (co)processors.[A uniform bus system operates without any special consideration by]

Summary Of The Invention

An object of the [programmer]present invention is to reduce the expense of wiring, and in particular to reduce the number of unit terminals. [There is]The present invention provides a uniform bus system which operates without any special consideration by the programmer. The present invention includes a permanent implementation of the bus system control. Memories and peripherals can be connected to the bus system without any special measures. Likewise, units can be cascaded with the help of the bus system.

[2. Description of the invention

2.1 Overview of the invention, Abstract]

The [invention describes]present invention in accordance with an exemplary embodiment provides a general bus system which combines a number of internal lines and leads them as a bundle to the terminals. The bus system control is predefined and does not require any influence by the programmer. Any number of memories, peripherals or other units can be connected to the bus system (e.g., for cascading).

Brief Description Of The [details and specific embodiments as well as features of the bus system according to this invention are the object of the patent claims]Drawings

Figure 1 illustrates a basic unit as a type A FPGA.

[2.2Detailed description of the invention

The following description encompasses several architectures which are conventionally controlled and configured by a primary logic unit,]Figure 2 illustrates a basic unit as [in DFPs, FPGAs, DPGAs, etc. Parts of the primary logic unit may be integrated on the unit]a type B FPGA.

re 3 illustrates a basic unit as a DFP.

Figure 4 illustrates line bundling in FPGAs in accordance with an exemplary embodiment of the present invention.

Figure 5 illustrates line bundling in DFPs in accordance with an exemplary embodiment of the present invention.

Figure 6 illustrates an OUTPUT CELL in accordance with an exemplary embodiment of the present invention.

Figure 7 illustrates an INPUT CELL in accordance with an exemplary embodiment of the present invention.

Figure 8 illustrates address generation in accordance with an exemplary embodiment of the present invention.

Figure 9 illustrates a complete bus system with controller in accordance with an exemplary embodiment of the present invention.

Figure 10 illustrates the connection of memories and peripherals in accordance with an exemplary embodiment of the present invention.

Figure 11 illustrates an E-bus control register in accordance with an exemplary embodiment of the present invention.

Figure 12 illustrates an embodiment of the present invention employing RAMBUS.

Detailed Description Of The Invention

The following description encompasses several architectures which may be controlled and configured by a primary logic unit, as in DFPs, FPGAs, DPGAs, etc. Parts of the primary logic unit may be integrated on the unit. As an alternative, there is the possibility (Figures 6, 7) of dynamically controlling or reconfiguring the architectures directly through the unit itself. The architectures may be implemented in a permanent form on the unit, or they may be created only by configuring and possibly combining multiple logic cells, i.e., configurable cells which fulfill simple logical or arithmetic functions according to their configuration (cf. DFP, FPGA, DPGA).

[2.2.1]Bundling internal lines

: To obtain appropriate bus architectures, a plurality of internal lines are combined in buses (I-BUSn, where n denotes the number of the bus). The lines may be internal bus systems or lines of the edge cells. For write access to the external bus (E-Bus) over clocked latches or registers (I-GATE-REG), the individual buses are connected to gates that function as switches to the E-BUS. Such a unit is called an OUTPUT CELL. Access to the E-BUS takes place in such a way that the individual latches are switched via the gates to the common E-BUS. There is always only one gate open. Each I-BUSn has a unique identification number (n: e.g., I-BUS1, I-BUS976, ...).

For read access, the incoming E-BUS is stored temporarily in clocked latches or registers (E-GATE-REG) and then distributed over the gates to the I-BUSn. Such a unit is called an INPUT CELL. Pick up from the E-BUS takes place in such a way that an E-BUS transfer is written into one

or more E-GATE-REGs. The E-GATE-REGs can then be switched either individually or together to their internal bus systems.

The read/write accesses can take place in any order. Under some circumstances, it is appropriate to subdivide the internal buses I-BUSn into two groups, the writing output buses IO-BUSn and the reading input buses II-BUSn.

[2.2.2]Address generation

: For most accesses to external units, it is necessary to generate addresses for selecting a unit or parts of a unit. The addresses may be permanent, i.e., they do not change (this is the case especially with peripheral addresses) or the addresses may change by (usually) fixed values with each access (this is the case especially with memory addresses). For generating the addresses, there are programmable counters for read accesses and programmable counters for write accesses. The counters are set at a base value by the PLU, which is the unit that configures the configurable units (DFPs, FPGAs, DPGAs, etc.) based on cell architecture. With each access to the gate, the counter is incremented or decremented by a value defined by the PLU, depending on the setting. Likewise, each counter can also be used as a register, which means that counting is not performed with each access, and the value set in the counter is unchanged. The value of the counter belonging to the gate is assigned as an address to each bus transfer. The counter is set by a setting register (MODE PLUREG) to which the PLU has write access.

[2.2.3]Masks and states

: Each gate is assigned a number of bits in MODE PLUREG

which is described below, indicating whether the gate is active or is skipped by the controller, i.e., is masked out (MASK). This means that the gate is skipped in running through all gates to connect to the respective bus system.

The following mask records are conceivable:

- always skip the INPUT/OUTPUT CELL,
- skip the INPUT/OUTPUT CELL only in writing,
- skip the INPUT/OUTPUT CELL only in reading if the E-BUS MASTER has not accessed the INPUT/OUTPUT CELL,
- never skip the INPUT/OUTPUT CELL.

Each gate is assigned a state register which may be designed as an RS flip-flop. This register indicates whether data have been written into the register belonging to the gate.

[2.2.4 MODE PLUREG]

MODE PLUREG: The MODE PLUREG can be written and read by the PLU. It serves to set the bus system.

One possible MODE PLUREG architecture from the standpoint of PLU is illustrated below:

Bit l-m	Bit k-l	Bit 2-k	Bit 1	Bit 0
Mask	Predefined value	Step length	0 = additive counting 1 = subtractive counting	0 = register 1 = counter
Masking	Settings for address generator			

[2.2.5]Description of the INPUT CELL

: A distinction is made according to whether data [go]goes from the E-BUS to the unit (the component required for this is called the INPUT CELL) or whether data [go]goes from the unit to the E-BUS (the component required for this is called an OUTPUT CELL).

An INPUT CELL may be designed as follows. A latch (I-GATE-REG) which is controlled either by the external E-BUS MASTER or the internal state machine serves as a buffer for the data received from the E-BUS. The clock pulse of the latch is sent to (for example) an RS flip-flop (SET-REG) which retains access to the I-GATE-REG. Downstream from the I-GATE-REG is a gate (I-GATE) which is controlled by the state machine. The data goes from the I-GATE-REG to the I(I)-BUSn via the I-GATE.

In addition, there is a programmable incrementer/decrementer in the INPUT CELL. It can be controlled by the state machine after each active read access to the E-BUS to increment or decrement an adjustable value. It can also serve as a simple register. This counter generates the addresses for bus accesses where the unit is E-BUS MASTER. The addresses are sent to the E-BUS via a gate (ADR-GATE). The ADR-REG is controlled by the state machine.

The E-BUS MASTER can poll the state of the SET-REG via another gate (STATE-GATE). Each INPUT CELL has a MODE PLUREG in which the PLU configures the counter and turns the INPUT CELL on or off (masks it).

[2.2.6]Description of the OUTPUT CELL

: An OUTPUT CELL may be configured as follows. A latch

(E-GATE-REG) which is controlled by the internal state machine provides buffer storage for the data obtained from the I-BUS.

In addition, there is a programmable incrementer/decrementer in the OUTPUT CELL. The clock signal of the latch is sent to (for example) an RS flip-flop (SET-REG) which retains access to the E-GATE-REG. It can be controlled by the state machine after each read access to the E-BUS to increment or decrement an selectable value. It can also function as a simple register. This counter generates the addresses for bus accesses in which the unit is E-BUS MASTER.

The data of the E-GATE-REG, the addresses and the state of the SET-REG are sent to the E-BUS via a gate (E-GATE) which is controlled either by the external E-BUS MASTER or the internal state machine. Each OUTPUT CELL has a MODE PLUREG in which the PLU configures the counter and turns the OUTPUT CELL on and off (masks it).

[2.2.7]Controlling the bus system

: At a higher level than the individual gates, address generators and masks there is a controller consisting of a simple, known state machine. Two operating modes are differentiated:

1.An active mode in which the state machine controls the internal bus (I-BUS) and the external bus (E-BUS). This mode is called E-BUS MASTER because the state machine has control of the E-BUS.

2.A passive mode in which the state machine controls only the internal bus (I-BUS). The E-BUS is controlled by another external unit. The state machine reacts in this mode to the requirements of the external E-BUS MASTER.

This mode of operation is called E-BUS SLAVE.

The controller manages the E-BUS protocol. The sequence differs according to whether the controller is functioning in E-BUS MASTER or E-BUS SLAVE mode. No bus protocol is described in this paper, because a number of known protocols can be implemented.

[2.2.8]E-BUS MASTER and E-BUS SLAVE, EB-REG

: The E-BUS control register (EB-REG) is provided to manage the data traffic on the E-BUS. It is connected in series with the gates and can be addressed and operated from the E-BUS. The data exchange can be regulated through the following records:

I-WRITE: indicates that the I-BUS is written completely into the INPUT/OUTPUT CELLs,
I-READ: indicates that the I-BUS has completely read the INPUT/OUTPUT CELLs,
E-WRITE: indicates that the E-BUS has been written completely into the INPUT/OUTPUT CELLs,
E-READ: indicates that the E-BUS has completely read the INPUT/OUTPUT CELLs.

The EB-REG is always active only on the side of the E-BUS SLAVE, and the E-BUS MASTER has read and write access to it.

- All I-... records are written by E-BUS SLAVE and read by E-BUS MASTER.
- All E-... records are written by E-BUS MASTER and read by E-BUS SLAVE.

An E-BUS SLAVE can request control of the E-BUS by setting the REQ MASTER bit in its EB-REG. If the E-BUS MASTER recognizes the REQ MASTER bit, it must relinquish

the bus control as soon as possible. It does this by setting the MASTER bit in the EB-REG of an E-BUS SLAVE. It then immediately switches the E-BUS to passive mode. The old E-BUS SLAVE becomes the new E-BUS MASTER, and the old E-BUS MASTER becomes the new E-BUS SLAVE. The new E-BUS MASTER assumes control of the E-BUS. To recognize the first E-BUS MASTER after a RESET of the system, there is a terminal on each unit which indicates by the preset polarity whether the unit is E-BUS MASTER or E-BUS SLAVE after a RESET. The MASTER record in the EB-REG can also be set and reset by the PLU. The PLU must be sure that there are no bus collisions on the EB-BUS and that no ongoing transfers are interrupted.

[2.2.9]E-BUS MASTER writes data to E-BUS SLAVE

: The E-BUS MASTER can write data to the E-BUS SLAVE as follows:

[→The]→ The data transfer begins when the state machine of the E-BUS MASTER selects an OUTPUT CELL that is not masked out.

[→The]→ Data has already been stored in the I-GATE REG, depending on the design of the state machine, or the data is stored now.

[→The]→ The gate is activated.

[→The]→ The valid read address is transferred to the bus.

[→The data goes to the E-BUS and is stored in the E-GATE REG of the E-BUS SLAVE.]

→ The data goes to the E-BUS and is stored in the E-GATE REG of the E-BUS SLAVE.

→ The SET-REG in the E-BUS SLAVE is thus activated.

→ The gate in the E-BUS MASTER is deactivated.

→ The address counter generates the address for the next access.

- The transfer is terminated for the E-BUS MASTER.

There are two possible embodiments of the E-BUS SLAVE for transferring data from the bus to the unit:

1. [The data gate is always open and the data goes directly from the E-GATE-REG to the I-BUSn.
2. The state machine recognizes that SET-REG is activated, and it activates the gate, so that SET-REG can be reset.]

1. The data gate is always open and the data goes directly from the E-GATE-REG to the I-BUSn.
2. The state machine recognizes that SET-REG is activated, and it activates the gate, so that SET-REG can be reset.

The E-BUS MASTER can notify the E-BUS SLAVE when a complete bus cycle is terminated (a bus cycle is defined as the transfer of multiple data strings to different E-GATE-REGs, where each E-GATE-REG may be addressed exactly once).

- The E-BUS MASTER sets the E-WRITE bit in the EB-REG of the E-BUS SLAVE at the end of a bus cycle.
- The E-BUS SLAVE can respond by polling the INPUT CELLs.
- When it has polled all the INPUT CELLs, it sets the I-READ bit in its EB-REG.
- It then resets E-WRITE and all the SET-REGs of the INPUT CELLS.
- The E-BUS MASTER can poll I-READ and begin a new bus cycle after its activation.
- I-READ is reset by E-WRITE being written or the first bus transfer.

The E-BUS SLAVE can analyze whether the INPUT CELLs can/must be read again on the basis of the status of the EB-REG or the individual SET-REGs of the INPUT CELLs.

[2.2.10]E-BUS MASTER reads data from E-BUS SLAVE–

: From the standpoint of the E-BUS MASTER, there are two basic methods of reading data from the E-BUS SLAVE:

1. [Method in which the E-BUS data goes directly to the I-BUS:]

Method in which the E-BUS data goes directly to the I-BUS:

- The data transfer begins with the state machine of the E-BUS MASTER selecting an INPUT CELL which is not masked out.
- The I-GATE and the ADR-GATE are activated.
  [→The]→ The valid read address is transferred to the bus.
- The I-GATE-REG is transparent, i.e., it allows the data through to the I-BUSn.
- The gate in the E-BUS MASTER is deactivated.
- The address counter generates the address for the next access.
- The transfer is terminated for the E-BUS MASTER.

2. [Method in which the E-BUS data is stored temporarily in the I-GATE-REG:]

Method in which the E-BUS data is stored temporarily in the I-GATE-REG:

- The data transfer begins with the state machine of the E-BUS MASTER selecting an INPUT CELL which is not masked out.
- The I-GATE and the ADR-GATE are activated.
  [→The]→ The valid read address is

transferred to the bus.

- I-GATE-REG stores the data.
- The gate in the E-BUS MASTER is deactivated.
- The address counter generates the address for the next access.
- The E-BUS transfer is terminated for the E-BUS MASTER.
- All INPUT CELLS involved in the E-BUS transfer, which can be ascertained on the basis of the masks in the MODE PLUREG or the state of the SET-REG, are run through and the data is transferred to the respective I-BUS.

For the E-BUS SLAVE, the access looks as follows:

- The gate is activated by the E-BUS.
- The data and the state of any SET-REG that may be present go to the E-BUS.
- The gate is deactivated.

The E-BUS MASTER can notify the E-BUS SLAVE when a complete bus cycle is terminated.

- To do so, at the end of a bus cycle, the E-BUS MASTER sets the E-READ bit in the EB-REG of the E-BUS SLAVE.
- E-BUS SLAVE can react by writing to the OUTPUT CELLs anew.
- When it has written to all the OUTPUT CELLs, it sets the I-WRITE bit in its EB-REG.
- In doing so, it resets E-READ and all the SET-REGs of the OUTPUT CELLS.
- The E-BUS MASTER can poll I-WRITE and begin a new bus cycle after its activation.
- I-WRITE is reset by writing E-READ or the first

bus transfer.

E-BUS SLAVE can evaluate on the basis of the state of the EB-REG or the individual SET-REGs of the OUTPUT CELLS whether the OUTPUT CELLs can/must be written anew.

[2.2.11]Connection of memories and peripherals, cascading

: In addition to cascading identical units (DFPs, FPGAs, DPGAs), memories and peripherals can also be connected as lower-level SLAVE units (SLAVE) to the bus system described here. Memories and peripherals as well as other units (DFPs, FPGAs) can be combined here. Each connected SLAVE analyzes the addresses on the bus and recognizes independently whether it has been addressed. In these modes, the unit addressing the memory or the peripheral, i.e., the SLAVE units, is the bus MASTER (MASTER), i.e., the unit controls the bus and the data transfer. The exception is intelligent peripheral units, such as SCSI controllers that can initiate and execute transfers independently and therefore are E-BUS MASTERs.

[2.2.12 Abstract]

Through the method described here, bus systems can be connected easily and efficiently to DFPs and FPGAs. Both memories and peripherals as well as other units of the types mentioned above can be connected over the bus systems.

The bus system need not be implemented exclusively in DFPs, FPGAs and DPGAs. Hybrid operation of this bus system with traditional unit terminal architectures is of course possible. Thus the advantages of the respective technique can be utilized optimally.

Other sequencing methods are also conceivable for the bus system described [here] herein.

Figure 1: Figure 1 shows an FPGA, where 0101 represents the internal bus systems, 0102 includes one or more FPGA cells. [However, they will not be detailed here because they are free embodiment options that do not depend on the basic principle described here.]

[3.Brief description of the diagrams

Figure 1:Drawing of a basic unit as a type A FPGA
Figure 2:Drawing of a basic unit as a type B FPGA
Figure 3:Drawing of a basic unit as a DFP
Figure 4:Line bundling in FPGAs
Figure 5:Line bundling in DFPs
Figure 6:An OUTPUT CELL
Figure 7:An INPUT CELL
Figure 8:Address generation
Figure 9:Complete bus system with controller
Figure 10: Connection of memories and peripherals
Figure 11: The EB-REG
Figure 12: Embodiment]

[4. Detailed description of the diagrams

Figure 1 shows a known FPGA, where 0101 represents the internal bus systems, 0102 includes one or more FPGA cells.] 0103 denotes subbuses which are a subset of 0101 and are connected to 0101 via switches (crossbars). 0103 can also manage internal data of 0102 that are not switched to 0101. The FPGA cells are arranged in a two-dimensional array. 0104 is an edge cell located at the edge of the array and is thus in direct proximity to the terminals at the edge of the unit.

Figure 2: Figure 2 shows another [known]FPGA. This embodiment does not work with bus systems like 0101 but instead mainly with next-neighbor connections (0201) which are direct connections from an FPGA cell (0203) to a neighboring cell. Nevertheless, there can be global bus systems (0202), but they are not very wide. The FPGA cells or a group of FPGA cells have a connection to 0202. The FPGA cells or a group of FPGA cells have a connection to 0202. The FPGA cells are arranged in a two-dimensional array. 0204 is an edge cell located at the edge of the array and thus is in close proximity to the terminals at the edge of the unit.

Figure 3: Figure 3 shows a [DFP according to]PACT02 DFP (i.e., DE 44 16 881 A1). The PAE cells (0303) are wired to the bus systems (0301) via a bus interface (0304) in accordance with the present invention. Bus systems 0301 can be wired together via a bus switch (0302). The PAE cells are arranged in a two-dimensional array. 0305 is an edge cell located on the edge of the array and is thus in close proximity to the terminals at the edge of the unit.

Figure 4a: Figure 4a shows an FPGA edge according to Figure 1. Outside the edge cells (0401) there are arranged a plurality of INPUT/OUTPUT CELLs (0402) which connect the internal bus systems (0403) individually or in groups to the E-BUS (0404). The number of INPUT/OUTPUT CELLs depends on their own width in relation to the width of the internal bus systems. 0405 is an EB-REG. 0406 is a state machine. A bus system (0407) by means of which the state machine controls the INPUT/OUTPUT CELLs goes from the state machine to the EB-REG and each individual INPUT/OUTPUT CELL. There can be several 0405s and 0406s by combining a set of 0402s into groups, each managed by an 0405 and 0406.

Figure 4b: Figure 4b shows an FPGA edge according to Figure 2. Several INPUT/OUTPUT CELLs (0412) are arranged outside the edge cells (0411), with individual CELLs or groups of CELLs connected to the E-BUS (0414) via the internal bus systems (0413) and the direct connections of the edge cells (0417). The number of INPUT/OUTPUT CELLs depends on their own width in relation to the width of the internal bus systems (0413) and the number of direct connections (0418). 0415 is an EB-REG. 0416 is a state machine. A bus system (0417) by means of which the state machine controls the INPUT/OUTPUT CELLs goes from the state machine to the EB-REG and each individual INPUT/OUTPUT CELL. There may be multiple 0415s and 0416s by combining a number of 0412s into groups, each managed by a 0415 and 0416.

Figure 5: Figure 5 shows a DFP edge according to Figure 3. Outside the edge cells (0501) are arranged several INPUT/OUTPUT CELLs (0502) which are connected individually or in groups to the E-BUS (0504) by the internal bus systems (0503). The number of INPUT/OUTPUT CELLs depends on their own width in relation to the width of the internal bus systems (0503). 0505 is an EB-REG. 0506 is a state machine. The state machine controls the INPUT/OUTPUT CELLs via a bus system (0507) which goes from the state machine to the EB-REG and each individual INPUT/OUTPUT CELL. There may be multiple 0505s and 0506s by combining a number of 0412s into groups, each managed by a 0505 and 0506.

Figure 6: Figure 6 shows an OUTPUT CELL 0601. Outside of 0601 there are the EB-REG (0602) and the state machine (0603) plus a gate (0604) which connects the state machine to the E-BUS (0605) if it is the E-BUS MASTER. Access to the EB-REG is possible via the E-BUS (0605), the I-BUS (0613) and the PLU bus (0609). In addition, when the unit is reset, the MASTER bit can be set via an

external terminal (0614) leading out of the unit. The state machine (0603) has read and write access to 0602. In the OUTPUT CELL there is a multiplexer (0606) which assigns control of the E-GATE (0607) to either the E-BUS MASTER or the state machine (0603). The MODE PLUREG (0608) is set via the PLU bus (0609) or the I-BUS (0613) and it configures the address counter (0610) and the state machine (e.g., masking out the OUTPUT CELL). If data of the I-BUS (0613) is stored in the I-GATE-REG (0611), the access is noted in SET-REG (0612). The state of 0612 can be polled via 0607 on the E-BUS. Read access (E-GATE 0607 is activated) resets 0612. The addresses generated by 0610 and the data of 0611 are transferred to the E-BUS via gate 0607. There is the possibility of dynamically reconfiguring and controlling the OUTPUT CELL via the unit itself (DFP, FPGA, DPGA, etc.) rather than through the PLU. The I-BUS connection to the EB-REG (0602) and the MODE PLUREG (0608) serves this function.

Figure 7: Figure 7 shows an INPUT CELL 0701. Outside of 0701 there are the EB-REG (0702) and the state machine (0703), as well as a gate (MASTER GATE) (0704) which connects the state machine to the E-BUS (0705) if it is in the E-BUS MASTER mode. Access to EB-REG is possible via the E-BUS (0705), the I-BUS (0713) and the PLU bus (0709). Furthermore, when the unit is reset, the MASTER bit can be set via an external terminal (0714) leading out of the unit. The state machine (0703) has read and write access to 0702. In the INPUT CELL there is a multiplexer (0706) which assigns control of the E-GATE-REG (0707) to either the E-BUS MASTER or the state machine (0703). The MODE PLUREG (0708) is set via the PLU bus (0709) or the I-BUS (0713) and it configures the address counter (0710) and the state machine (e.g., masking out the INPUT CELL). If data of the E-BUS (0705) is stored in the E-GATE-REG (0707), the access is noted in the SET-REG (0712). The state of 0712 can be polled on

the E-BUS via a gate (0715) whose control is the same as that of the latch (0707). A read access - E-GATE 0711 is activated and the data goes to the I-BUS (0713) - resets 0712 via 0717. As an alternative, 0712 can be reset (0718) via the state machine (0703). The addresses generated by 0710 are transferred via the gate (ADR-GATE) 0716 to the E-BUS. 0716 is activated by the state machine (0703) when it is E-BUS MASTER. There is the possibility of dynamically reconfiguring and controlling the INPUT CELL via the unit itself (DFP, FPGA, DPGA, etc.) instead of through the PLU. The I-BUS connection to the EB-REG (0702) and the MODE PLUREG (0708) serves this function.

Figure 8: Figure 8 shows the MODE PLUREG of an INPUT or OUTPUT CELL written by the ~~PLU via the~~ PLU via the PLU bus (0802) or via an I-BUS (0808). The respective bus system is selected by the multiplexer (0809) (control of the multiplexer is not shown because an ordinary decoder logic can be used). The counter settings such as step length, counting direction and enabling of the counter are sent directly (0807) to the counter (0803). The basic address can either be written directly (0805) into the counter via a load (0804) or stored temporarily in an extension (0811) of 0801. Records in 0801 that are relevant for the state machine go to the state machine via a gate (0806) which is opened by the state machine for the INPUT or OUTPUT CELL activated at the time.

Figure 9a: Figure 9a shows a bus interface circuit with a state machine (0901), MASTER GATE (0902) and EB-REG (0903). The INPUT CELLs (0904) transfer data from the E-BUS (0905) to the II-BUS (0906). The OUTPUT CELLs (0907) transfer data from the IO-BUS (0908) to the E-BUS (0905). All units are linked together by the control bus (0909).

Figure 9b: Figure 9b shows a bus interface circuit with state machine (0901), MASTER GATE (0902) and EB-REG

(0903). The INPUT CELLs (0904) transfer data from the E-BUS (0905) to the bidirectional I-BUS (0910). The OUTPUT CELLS (0907) transfer data from the bidirectional I-BUS (0910) to the E-BUS (0905). All units are linked together over the control bus (0909). Interface circuits which use both possibilities (Figures 9a and 9b) in a hybrid design are also conceivable.

Figure 10a: Figure 10a shows the interconnections of two units (DFPs, FPGAs, DPGAs, etc.) (1001) interconnected via the E-BUS (1002).

Figure 10b: Figure 10b shows the interconnection of a number of units (DFPs, FPGAs, DPGAs, etc.) (1001) via the E-BUS (1002).

Figure 10c: Figure 10c shows the interconnection of a number of units (DFPs, FPGAs, DPGAs, etc.) (1001) via the E-BUS (1002). The interconnection can be expanded to a matrix. One unit (1001) may also manage multiple bus systems (1002).

Figure 10d: Figure 10d shows the interconnection of a unit (DFP, FPGA, DPGA, etc.) (1001) to a memory unit or a memory bank (1003) via the E-BUS (1002).

Figure 10e: Figure 10e shows the interconnection of a unit (DFP, FPGA, DPGA, etc.) (1001) to a peripheral device or a peripheral group (1004) via the E-BUS (1002).

Figure 10f: Figure 10f shows the interconnection of a unit (DFP, FPGA, DPGA, etc.) (1001) to a memory unit or a memory bank (1003) and to a peripheral device or a peripheral group (1004) via the E-BUS (1002).

Figure 10g: Figure 10g shows the interconnection of a

unit (DFP, FPGA, DPGA, etc.) (1001) to a memory unit or a memory bank (1003) and to a peripheral device or a peripheral group (1004) plus another unit (DFP, FPGA, DPGA, etc.) (1001) via the E-BUS (1002).

Figure 11: Figure 11 shows the architecture of the EB-REG. The bus systems E-BUS (1103), the PLU bus (1104) over which the PLU has access to the EB-REG, and the local internal bus between the INPUT/OUTPUT CELLs, the state machine and the EB-REG (1105, see 0407, 0417, 0517) and possibly an I-BUS (1114) are connected to a multiplexer (1106). The multiplexer (1106) selects either one of the buses or the feedback to the register (1108) and switches the data through to the input of the register (1108). The MASTER bit is sent separately over the multiplexer (1107) to the register (1108). The multiplexer is controlled by the RESET signal (1101) (resetting or initializing the unit). If a RESET signal is applied, the multiplexer (1107) switches the signal of an external chip connection (1102) through to the input of the register (1108); otherwise the output of the multiplexer (1106) is switched through to the input of the register (1108). Thus MASTER may be preallocated. The register (1108) is clocked by the system clock (1112). The contents of the register (1108) are switched via a gate (1109, 1110, 1111, 1113) to the respective bus system (1103, 1104, 1105, 1114) having read access at that time. The control of the gates (1109, 1110, 1111, 1113) and of the multiplexer (1106) is not shown because an ordinary decoder logic may be used.

~~5. Embodiments~~

: Figure 12 shows an example using the standard bus system RAMBUS (1203). A unit (DFP, FPGA, DPGA, etc.) (1201) is connected to other units (memories,

peripherals, other DFPs, FPGAs, DPGAs, etc.) (1202) over the bus system (1203). Independently of the bus system (1203), the unit (1201) may have additional connecting lines (1204), e.g., as is customary in the related art, for connecting any desired circuits.

Definition of terms: The following is a definition of terms used above.

ADR-GATE: Gate which switches addresses to the E-BUS if the unit is in E-BUS MASTER mode.

[6.Definition of terms]

[ADR-GATE: Gate which switches addresses to the E-BUS if the unit is in E-BUS MASTER mode.]

DFP: Data flow processor according to German Patent DE 44 16 881.

DPGA: Dynamically programmable gate array. Related art.

D flip-flop: Storage element which stores a signal at the rising edge of a clock pulse.

EB-REG: Register that stores the status signals between I-BUS and E-BUS.

E-BUS: External bus system outside a unit.

E-BUS MASTER: Unit that controls the E-BUS. Active.

E-BUS SLAVE: Unit controlled by the E-BUS MASTER. Passive.

E-GATE: Gate which is controlled by the internal state machine of the unit or by the E-BUS MASTER and switches

data to the E-BUS.

E-GATE-REG: Register into which data transmitted to the E-BUS over the E-GATE is entered.

E-READ: Flag in the EB-REG indicating that the OUTPUT CELLS have been transferred completely to the E-BUS.

E-WRITE: Flag in the EB-REG indicating that the E-BUS has been transferred completely to the INPUT CELLS.

Flag: Status bit in a register, indicating a state.

FPGA: Field programmable gate array. Related art.

Handshake: Signal protocol where a signal A indicates a state and another signal B confirms that it has accepted signal A and responded to it.

INPUT CELL: Unit transmitting data from the E-BUS to an I-BUS.

I-BUSn *(also I-BUS)*: Internal bus system of a unit, which may also consist of bundles of individual lines, where n indicates the number of the bus.

II-BUSn *(also II-BUS)*: Internal bus system of a unit, which may also consist of bundles of individual lines, with n denoting the number of the bus. The bus is driven by an INPUT CELL and goes to logic inputs.

IO-BUSn *(also IO-BUS)*: Internal bus system of a unit, which may also consist of bundles of individual lines, with n denoting the number of the bus. The bus is driven by logic outputs and goes to an OUTPUT CELL. n indicates the number of the bus.

I-GATE: Gate that switches data to the I-BUS.

I-GATE-REG: Register which is controlled by the internal state machine or by E-BUS MASTER and into which data transmitted over the I-GATE to the I-BUS is entered.

I-READ: Flag in the EB-REG indicating that the INPUT CELLs have been completely transferred to the I-BUS.

I-WRITE: Flag in the EB-REG indicating that the I-BUS has been completely transferred to the OUTPUT CELLs.

Edge cell: Cell at the edge of a cell array, often with direct contact with the terminals of a unit.

Configuring: Setting the function and interconnecting a logic unit, a (FPGA) cell (logic cell) or a PAE (see reconfiguring).

Primary logic unit (PLU): Unit for configuring and reconfiguring a PAE or logic cell. Embodied by a microcontroller specifically designed for this purpose.

Latch: Storage element which usually relays a signal transparently during the H level and stores it during the L level. Latches where the function of levels is exactly the opposite are sometimes used in PAEs. An inverter is then connected before the clock pulse of a conventional latch.

Logic cells: Configurable cells used in DFPs, FPGAs, DPGAs, fulfilling simple logical or arithmetic functions, depending on configuration.

MASTER: Flag in EB-REG showing that the E-BUS unit is a MASTER.

MODE PLUREG: Register in which the primary logic unit sets the configuration of an INPUT/OUTPUT CELL.

OUTPUT CELL: Unit that transmits data from an I-BUS to the E-BUS.

PAE: Processing array element: EALU with O-REG, R-REG, R20-MUX, F-PLUREG, M-PLUREG, BM UNIT, SM UNIT, sync UNIT, state-back UNIT and power UNIT.

PLU: Unit for configuring and reconfiguring a PAE or a logic cell. Embodied by a microcontroller specifically designed for this purpose.

REQ-MASTER: Flag in the EB-REG indicating that the unit would like to become E-BUS MASTER.

RS flip-flop: Reset/set flip-flop. Storage element which can be switched by two signals.

SET-REG: Register indicating that data has been written in an I-GATE-REG or E-GATE-REG but not yet read.

STATE-GATE: Gate switching the output of the SET-REG to the E-BUS.

Gate: Switch that relays or blocks a signal. Simple comparison: relay.

Reconfiguring: New configuration of any number of PAEs or logic cells while any remaining number of PAEs or logic cells continue their own function (see configuring).

State machine: Logic which can assume miscellaneous states. The transitions between states depend on various input parameters. These machines are used to control complex functions and belong to the related art.

[7.]Naming conventions:

Unit: -UNIT
Operating mode: -MODE
Multiplexer: -MUX
Negated signal: not-
Register for PLU visible: -PLUREG
Internal register: -REG
Shift registers: -sft

Function convention:

Shift registers: sft

AND function: &

[Conventions]

[7.1.Naming conventions

Unit: -UNIT
Operating mode: -MODE
Multiplexer: -MUX
Negated signal: not-
Register for PLU visible: -PLUREG
Internal register: -REG
Shift registers: -sft]

[7.2Function convention

Shift registers: sft

AND function: &]

A	B	Q
0	0	0
0	1	[10000QBA]0
1	0	0
1	1	1

OR function: #

A	B	[Q]Q
[0]0	0	0
0	1	[1]1
[1]1	[0]0	1
1	1	1

NOT function: !

A	Q
0	1
1	0

GATE function: G

EN	D	Q
0	0	-
0	1	-
1	0	0
1	1	1

ABSTRACT

A uniform bus system is provided which operates without any special consideration by a programmer. Memories and peripheral may be connected to this bus system without any special measures. Likewise, units may be cascaded with the help of the bus system. The bus system combines a number of internal lines, and leads them as a bundle to terminals. The bus system control is predefined and does not require any influence by the programmer. Any number of memories, peripherals or other units can be connected to the bus system.

I/O and Memory Bus System for DFPs and Units with Two- or Multi-Dimensional Programmable Cell Architectures

Field Of The Invention

The present invention relates to I/O and memory bus systems.

Background Information

~~As described in~~ In German Patent No. DE 44 16 881 A1, a data flow processor ("DFP") is described in which the lines of each edge cell (i.e., a cell at the edge of a cell array, often in direct contact with the terminals of the unit) are guided out via the terminals of the unit. The lines do not have any specific function, and instead they assume the function implemented in the edge cells. If several DFPs are interconnected, all terminals are connected to form a matrix.

In systems with two- or multi-dimensional programmable cell architectures (FPGAs, DPGAs), a certain subset of internal bus systems and lines of the edge cells are connected to the outside via the unit terminals. The lines do not have any specific function, and instead they assume the function written in the edge cells. If several FPGAs/DPGAs are interconnected, the terminals assume the function implemented in the hardware or software.

The wiring expense for the periphery or for interconnecting DFPs is very high, because the programmer must ensure that the respective functions are integrated into the cells of the DFP(s) at the same time. For connecting a memory, a memory management unit must be integrated into the unit. Connection of peripherals must also be supported. Additionally, the cascading of DFPs must be similarly taken into account. Not only is the

expense relatively high, but also area on the unit is lost for the respective implementations. This also applies to FPGAs and DPGAs, particularly when they are used for implementation of algorithms and when they work as arithmetic (co)processors.

Summary Of The Invention

An object of the present invention is to reduce the expense of wiring, and in particular to reduce the number of unit terminals. The present invention provides a uniform bus system which operates without any special consideration by the programmer. The present invention includes a permanent implementation of the bus system control. Memories and peripherals can be connected to the bus system without any special measures. Likewise, units can be cascaded with the help of the bus system.

The present invention in accordance with an exemplary embodiment provides a general bus system which combines a number of internal lines, and leads them as a bundle to the terminals. The bus system control is predefined and does not require any influence by the programmer. Any number of memories, peripherals or other units can be connected to the bus system (e.g., for cascading).

Brief Description Of The Drawings

Figure 1 illustrates a basic unit as a type A FPGA.

Figure 2 illustrates a basic unit as a type B FPGA.

Figure 3 illustrates a basic unit as a DFP.

Figures 4a and 4b illustrate ~~Figure 4 illustrates~~ line bundling in FPGAs in accordance with an exemplary embodiment of the present invention.

Figure 5 illustrates line bundling in DFPs in accordance with an exemplary embodiment of the present invention.

Figure 6 illustrates an OUTPUT CELL in accordance with an exemplary embodiment of the present invention.

Figure 7 illustrates an INPUT CELL in accordance with an exemplary embodiment of the present invention.

Figure 8 illustrates address generation in accordance with an exemplary embodiment of the present invention.

B Figures 9a-9b illustrate ~~Figure 9 illustrates~~ a complete bus system with controller in accordance with an exemplary embodiment of the present invention.

B Figures 10a-10g illustrate ~~Figure 10 illustrates~~ the connection of memories and peripherals in accordance with an exemplary embodiment of the present invention.

Figure 11 illustrates an E-bus control register in accordance with an exemplary embodiment of the present invention.

Figure 12 illustrates an embodiment of the present invention employing RAMBUS.

Detailed Description Of The Invention

The following description encompasses several architectures which may be controlled and configured by a primary logic unit, as in DFPs, FPGAs, DPGAs, etc. Parts of the primary logic unit may be integrated on the unit. As an alternative, there is the possibility (Figures 6, 7) of dynamically controlling or reconfiguring the architectures directly through the unit itself. The architectures may be implemented in a permanent form on the unit, or they may be created only by configuring and possibly combining multiple logic cells, i.e., configurable cells which fulfill simple logical or arithmetic functions according to their configuration

4

(cf. DFP, FPGA, DPGA).

Bundling internal lines: To obtain appropriate bus architectures, a plurality of internal lines are combined in buses (I-BUSn, where n denotes the number of the bus). The lines may be internal bus systems or lines of the edge cells. For write access to the external bus (E-Bus) over clocked latches or registers (I-GATE-REG), the individual buses are connected to gates that function as switches to the E-BUS. Such a unit is called an OUTPUT CELL. Access to the E-BUS takes place in such a way that the individual latches are switched via the gates to the common E-BUS. There is always only one gate open. Each I-BUSn has a unique identification number (n: e.g., I-BUS1, I-BUS976, ...).

For read access, the incoming E-BUS is stored temporarily in clocked latches or registers (E-GATE-REG) and then distributed over the gates to the I-BUSn. Such a unit is called an INPUT CELL. Pick up from the E-BUS takes place in such a way that an E-BUS transfer is written into one or more E-GATE-REGs. The E-GATE-REGs can then be switched either individually or together to their internal bus systems.

The read/write accesses can take place in any order. Under some circumstances, it is appropriate to subdivide the internal buses I-BUSn into two groups, the writing output buses IO-BUSn and the reading input buses II-BUSn.

Address generation: For most accesses to external units, it is necessary to generate addresses for selecting a unit or parts of a unit. The addresses may be permanent, i.e., they do not change (this is the case especially with peripheral addresses) or the addresses may change by (usually) fixed values with each access (this is the case especially with memory addresses). For generating the

addresses, there are programmable counters for read accesses and programmable counters for write accesses. The counters are set at a base value by the PLU, which is the unit that configures the configurable units (DFPs, FPGAs, DPGAs, etc.) based on cell architecture. With each access to the gate, the counter is incremented or decremented by a value defined by the PLU, depending on the setting. Likewise, each counter can also be used as a register, which means that counting is not performed with each access, and the value set in the counter is unchanged. The value of the counter belonging to the gate is assigned as an address to each bus transfer. The counter is set by a setting register (MODE PLUREG) to which the PLU has write access.

Masks and states: Each gate is assigned a number of bits in MODE PLUREG which is described below, indicating whether the gate is active or is skipped by the controller, i.e., is masked out (MASK). This means that the gate is skipped in running through all gates to connect to the respective bus system.

The following mask records are conceivable:

- always skip the INPUT/OUTPUT CELL,
- skip the INPUT/OUTPUT CELL only in writing,
- skip the INPUT/OUTPUT CELL only in reading if the E-BUS MASTER has not accessed the INPUT/OUTPUT CELL,
- never skip the INPUT/OUTPUT CELL.

Each gate is assigned a state register which may be designed as an RS flip-flop. This register indicates whether data have been written into the register belonging to the gate.

MODE PLUREG: The MODE PLUREG can be written and read by the PLU. It serves to set the bus system.

6

One possible MODE PLUREG architecture from the standpoint of PLU is illustrated below:

Bit l-m	Bit k-l	Bit 2-k	Bit 1	Bit 0
Mask	Predefined value	Step length	0 = additive counting 1 = subtractive counting	0 = register 1 = counter
Masking	Settings for address generator			

Description of the INPUT CELL: A distinction is made according to whether data goes from the E-BUS to the unit (the component required for this is called the INPUT CELL) or whether data goes from the unit to the E-BUS (the component required for this is called an OUTPUT CELL).

An INPUT CELL may be designed as follows. A latch (I-GATE-REG) which is controlled either by the external E-BUS MASTER or the internal state machine serves as a buffer for the data received from the E-BUS. The clock pulse of the latch is sent to (for example) an RS flip-flop (SET-REG) which retains access to the I-GATE-REG. Downstream from the I-GATE-REG is a gate (I-GATE) which is controlled by the state machine. The data goes from the I-GATE-REG to the I(I)-BUSn via the I-GATE.

In addition, there is a programmable incrementer/decrementer in the INPUT CELL. It can be controlled by the state machine after each active read access to the E-BUS to increment or decrement an adjustable value. It can also serve as a simple register. This counter generates the addresses for bus accesses where the unit is E-BUS MASTER. The addresses are sent to

the E-BUS via a gate (ADR-GATE). The ADR-REG is controlled by the state machine.

The E-BUS MASTER can poll the state of the SET-REG via another gate (STATE-GATE). Each INPUT CELL has a MODE PLUREG in which the PLU configures the counter and turns the INPUT CELL on or off (masks it).

Description of the OUTPUT CELL: An OUTPUT CELL may be configured as follows. A latch (E-GATE-REG) which is controlled by the internal state machine provides buffer storage for the data obtained from the I-BUS.

In addition, there is a programmable incrementer/decrementer in the OUTPUT CELL. The clock signal of the latch is sent to (for example) an RS flip-flop (SET-REG) which retains access to the E-GATE-REG. It can be controlled by the state machine after each read access to the E-BUS to increment or decrement an selectable value. It can also function as a simple register. This counter generates the addresses for bus accesses in which the unit is E-BUS MASTER.

The data of the E-GATE-REG, the addresses and the state of the SET-REG are sent to the E-BUS via a gate (E-GATE) which is controlled either by the external E-BUS MASTER or the internal state machine. Each OUTPUT CELL has a MODE PLUREG in which the PLU configures the counter and turns the OUTPUT CELL on and off (masks it).

Controlling the bus system: At a higher level than the individual gates, address generators and masks there is a controller consisting of a simple, known state machine. Two operating modes are differentiated:

1. An active mode in which the state machine controls the internal bus (I-BUS) and the external bus (E-BUS). This

mode is called E-BUS MASTER because the state machine has control of the E-BUS.

2.A passive mode in which the state machine controls only the internal bus (I-BUS). The E-BUS is controlled by another external unit. The state machine reacts in this mode to the requirements of the external E-BUS MASTER. This mode of operation is called E-BUS SLAVE.

The controller manages the E-BUS protocol. The sequence differs according to whether the controller is functioning in E-BUS MASTER or E-BUS SLAVE mode. No bus protocol is described in this paper, because a number of known protocols can be implemented.

E-BUS MASTER and E-BUS SLAVE, EB-REG: The E-BUS control register (EB-REG) is provided to manage the data traffic on the E-BUS. It is connected in series with the gates and can be addressed and operated from the E-BUS. The data exchange can be regulated through the following records:

I-WRITE: indicates that the I-BUS is written completely into the INPUT/OUTPUT CELLs,
I-READ: indicates that the I-BUS has completely read the INPUT/OUTPUT CELLs,
E-WRITE: indicates that the E-BUS has been written completely into the INPUT/OUTPUT CELLs,
E-READ: indicates that the E-BUS has completely read the INPUT/OUTPUT CELLs.

The EB-REG is always active only on the side of the E-BUS SLAVE, and the E-BUS MASTER has read and write access to it.

- All I-.. records are written by the E-BUS SLAVE and read by the E-BUS MASTER.

- All E-. records are written by E-BUS MASTER and read by E-BUS SLAVE.

An E-BUS SLAVE can request control of the E-BUS by setting the REQ MASTER bit in its EB-REG. If the E-BUS MASTER recognizes the REQ MASTER bit, it must relinquish the bus control as soon as possible. It does this by setting the MASTER bit in the EB-REG of an E-BUS SLAVE. It then immediately switches the E-BUS to passive mode. The old E-BUS SLAVE becomes the new E-BUS MASTER, and the old E-BUS MASTER becomes the new E-BUS SLAVE. The new E-BUS MASTER assumes control of the E-BUS. To recognize the first E-BUS MASTER after a RESET of the system, there is a terminal on each unit which indicates by the preset polarity whether the unit is E-BUS MASTER or E-BUS SLAVE after a RESET. The MASTER record in the EB-REG can also be set and reset by the PLU. The PLU must be sure that there are no bus collisions on the EB-BUS and that no ongoing transfers are interrupted.

E-BUS MASTER writes data to E-BUS SLAVE: The E-BUS MASTER can write data to the E-BUS SLAVE as follows:

- The data transfer begins when the state machine of the E-BUS MASTER selects an OUTPUT CELL that is not masked out.
- Data has already been stored in the I-GATE REG, depending on the design of the state machine, or the data is stored now.
- The gate is activated.
  - The valid read address is transferred to the bus.
  - The data goes to the E-BUS and is stored in the E-GATE REG of the E-BUS SLAVE.
- The SET-REG in the E-BUS SLAVE is thus activated.
- The gate in the E-BUS MASTER is deactivated.
- The address counter generates the address for the

10

next access.

- The transfer is terminated for the E-BUS MASTER.

There are two possible embodiments of the E-BUS SLAVE for transferring data from the bus to the unit:

1. The data gate is always open and the data goes directly from the E-GATE-REG to the I-BUSn.
2. The state machine recognizes that SET-REG is activated, and it activates the gate, so that SET-REG can be reset.

The E-BUS MASTER can notify the E-BUS SLAVE when a complete bus cycle is terminated (a bus cycle is defined as the transfer of multiple data strings to different E-GATE-REGs, where each E-GATE-REG may be addressed exactly once).

- The E-BUS MASTER sets the E-WRITE bit in the EB-REG of the E-BUS SLAVE at the end of a bus cycle.
- The E-BUS SLAVE can respond by polling the INPUT CELLs.
- When it has polled all the INPUT CELLs, it sets the I-READ bit in its EB-REG.
- It then resets E-WRITE and all the SET-REGs of the INPUT CELLS.
- The E-BUS MASTER can poll I-READ and begin a new bus cycle after its activation.
- I-READ is reset by E-WRITE being written or the first bus transfer.

The E-BUS SLAVE can analyze whether the INPUT CELLs can/must be read again on the basis of the status of the EB-REG or the individual SET-REGs of the INPUT CELLs.

E-BUS MASTER reads data from E-BUS SLAVE: From the standpoint of the E-BUS MASTER, there are two basic

11

methods of reading data from the E-BUS SLAVE:

1. Method in which the E-BUS data goes directly to the I-BUS:

- → The data transfer begins with the state machine of the E-BUS MASTER selecting an INPUT CELL which is not masked out.
- → The I-GATE and the ADR-GATE are activated.
  - → The valid read address is transferred to the bus.
- → The I-GATE-REG is transparent, i.e., it allows the data through to the I-BUSn.
- → The gate in the E-BUS MASTER is deactivated.
- → The address counter generates the address for the next access.
- → The transfer is terminated for the E-BUS MASTER.

2. Method in which the E-BUS data is stored temporarily in the I-GATE-REG:

- → The data transfer begins with the state machine of the E-BUS MASTER selecting an INPUT CELL which is not masked out.
- → The I-GATE and the ADR-GATE are activated.
  - → The valid read address is transferred to the bus.
- → I-GATE-REG stores the data.
- → The gate in the E-BUS MASTER is deactivated.
- → The address counter generates the address for the next access.
- → The E-BUS transfer is terminated for the E-BUS MASTER.
- → All INPUT CELLS involved in the E-BUS

transfer, which can be ascertained on the basis of the masks in the MODE PLUREG or the state of the SET-REG, are run through and the data is transferred to the respective I-BUS.

B For the E-BUS SLAVE, the access may be accomplished ~~looks~~ as follows:

- The gate is activated by the E-BUS.
- The data and the state of any SET-REG that may be present go to the E-BUS.
- The gate is deactivated.

The E-BUS MASTER can notify the E-BUS SLAVE when a complete bus cycle is terminated.

- To do so, at the end of a bus cycle, the E-BUS MASTER sets the E-READ bit in the EB-REG of the E-BUS SLAVE.
- E-BUS SLAVE can react by writing to the OUTPUT CELLs anew.
- When it has written to all the OUTPUT CELLs, it sets the I-WRITE bit in its EB-REG.
- In doing so, it resets E-READ and all the SET-REGs of the OUTPUT CELLS.
- The E-BUS MASTER can poll I-WRITE and begin a new bus cycle after its activation.
- I-WRITE is reset by writing E-READ or the first bus transfer.

B The E-BUS SLAVE can evaluate on the basis of the state of the EB-REG or the individual SET-REGs of the OUTPUT CELLS whether the OUTPUT CELLs can/must be written anew.

Connection of memories and peripherals, cascading: In addition to cascading identical units (DFPs, FPGAs, DPGAs), memories and peripherals can also be connected as

lower-level SLAVE units (SLAVE) to the bus system described here. Memories and peripherals as well as other units (DFPs, FPGAs) can be combined here. Each connected SLAVE analyzes the addresses on the bus and recognizes independently whether it has been addressed. In these modes, the unit addressing the memory or the peripheral, i.e., the SLAVE units, is the bus MASTER (MASTER), i.e., the unit controls the bus and the data transfer. The exception is intelligent peripheral units, such as SCSI controllers that can initiate and execute transfers independently and therefore are E-BUS MASTERs.

Through the method described here, bus systems can be connected easily and efficiently to DFPs and FPGAs. Both memories and peripherals as well as other units of the types mentioned above can be connected over the bus systems.

The bus system need not be implemented exclusively in DFPs, FPGAs and DPGAs. Hybrid operation of this bus system with traditional unit terminal architectures is of course possible. Thus the advantages of the respective technique can be utilized optimally.

Other sequencing methods are also conceivable for the bus system described herein.

Figure 1: Figure 1 shows an FPGA, where 0101 represents the internal bus systems, 0102 includes one or more FPGA cells. 0103 denotes subbuses which are a subset of 0101 and are connected to 0101 via switches (crossbars). 0103 can also manage internal data of 0102 that are not switched to 0101. The FPGA cells are arranged in a two-dimensional array. 0104 is an edge cell located at the edge of the array and is thus in direct proximity to the terminals at the edge of the unit.

Figure 2: Figure 2 shows another FPGA. This embodiment does not work with bus systems like 0101 but instead mainly with next-neighbor connections (0201) which are direct connections from an FPGA cell (0203) to a neighboring cell. Nevertheless, there can be global bus systems (0202), but they are not very wide. The FPGA cells or a group of FPGA cells have a connection to 0202. The FPGA cells or a group of FPGA cells have a connection to 0202. The FPGA cells are arranged in a two-dimensional array. 0204 is an edge cell located at the edge of the array and thus is in close proximity to the terminals at the edge of the unit.

Figure 3: Figure 3 shows a PACT02 DFP (i.e., DE 44 16 881 A1). The PAE cells (0303) are wired to the bus systems (0301) via a bus interface (0304) in accordance with the present invention. Bus systems 0301 can be wired together via a bus switch (0302). The PAE cells are arranged in a two-dimensional array. 0305 is an edge cell located on the edge of the array and is thus in close proximity to the terminals at the edge of the unit.

Figure 4a: Figure 4a shows an FPGA edge according to Figure 1. Outside the edge cells (0401) there are arranged a plurality of INPUT/OUTPUT CELLs (0402) which connect the internal bus systems (0403) individually or in groups to the E-BUS (0404). The number of INPUT/OUTPUT CELLs depends on their own width in relation to the width of the internal bus systems. 0405 is an EB-REG. 0406 is a state machine. A bus system (0407) by means of which the state machine controls the INPUT/OUTPUT CELLs goes from the state machine to the EB-REG and each individual INPUT/OUTPUT CELL. There can be several 0405s and 0406s by combining a set of 0402s into groups, each managed by an 0405 and 0406.

Figure 4b: Figure 4b shows an FPGA edge according to Figure 2. Several INPUT/OUTPUT CELLs (0412) are arranged outside the edge cells (0411), with individual CELLs or groups of CELLs connected to the E-BUS (0414) via the internal bus systems (0413) and the direct connections of the edge cells (0417). The number of INPUT/OUTPUT CELLs depends on their own width in relation to the width of the internal bus systems (0413) and the number of direct connections (0418). 0415 is an EB-REG. 0416 is a state machine. A bus system (0417) by means of which the state machine controls the INPUT/OUTPUT CELLs goes from the state machine to the EB-REG and each individual INPUT/OUTPUT CELL. There may be multiple 0415s and 0416s by combining a number of 0412s into groups, each managed by a 0415 and 0416.

Figure 5: Figure 5 shows a DFP edge according to Figure 3. Outside the edge cells (0501) are arranged several INPUT/OUTPUT CELLs (0502) which are connected individually or in groups to the E-BUS (0504) by the internal bus systems (0503). The number of INPUT/OUTPUT CELLs depends on their own width in relation to the width of the internal bus systems (0503). 0505 is an EB-REG. 0506 is a state machine. The state machine controls the INPUT/OUTPUT CELLs via a bus system (0507) which goes from the state machine to the EB-REG and each individual INPUT/OUTPUT CELL. There may be multiple 0505s and 0506s by combining a number of 0412s into groups, each managed by a 0505 and 0506.

Figure 6: Figure 6 shows an OUTPUT CELL 0601. Outside of 0601 there are the EB-REG (0602) and the state machine (0603) plus a gate (0604) which connects the state machine to the E-BUS (0605) if it is the E-BUS MASTER. Access to the EB-REG is possible via the E-BUS (0605), the I-BUS (0613) and the PLU bus (0609). In addition, when the unit is reset, the MASTER bit can be set via an

external terminal (0614) leading out of the unit. The state machine (0603) has read and write access to 0602. In the OUTPUT CELL there is a multiplexer (0606) which assigns control of the E-GATE (0607) to either the E-BUS MASTER or the state machine (0603). The MODE PLUREG (0608) is set via the PLU bus (0609) or the I-BUS (0613) and it configures the address counter (0610) and the state machine (e.g., masking out the OUTPUT CELL). If data of the I-BUS (0613) is stored in the I-GATE-REG (0611), the access is noted in SET-REG (0612). The state of 0612 can be polled via 0607 on the E-BUS. Read access (E-GATE 0607 is activated) resets 0612. The addresses generated by 0610 and the data of 0611 are transferred to the E-BUS via gate 0607. There is the possibility of dynamically reconfiguring and controlling the OUTPUT CELL via the unit itself (DFP, FPGA, DPGA, etc.) rather than through the PLU. The I-BUS connection to the EB-REG (0602) and the MODE PLUREG (0608) serves this function.

Figure 7: Figure 7 shows an INPUT CELL 0701. Outside of 0701 there are the EB-REG (0702) and the state machine (0703), as well as a gate (MASTER GATE) (0704) which connects the state machine to the E-BUS (0705) if it is in the E-BUS MASTER mode. Access to EB-REG is possible via the E-BUS (0705), the I-BUS (0713) and the PLU bus (0709). Furthermore, when the unit is reset, the MASTER bit can be set via an external terminal (0714) leading out of the unit. The state machine (0703) has read and write access to 0702. In the INPUT CELL there is a multiplexer (0706) which assigns control of the E-GATE-REG (0707) to either the E-BUS MASTER or the state machine (0703). The MODE PLUREG (0708) is set via the PLU bus (0709) or the I-BUS (0713) and it configures the address counter (0710) and the state machine (e.g., masking out the INPUT CELL). If data of the E-BUS (0705) is stored in the E-GATE-REG (0707), the access is noted in the SET-REG (0712). The state of 0712 can be polled on

17

the E-BUS via a gate (0715) whose control is the same as that of the latch (0707). A read access - E-GATE 0711 is activated and the data goes to the I-BUS (0713) - resets 0712 via 0717. As an alternative, 0712 can be reset (0718) via the state machine (0703). The addresses generated by 0710 are transferred via the gate (ADR-GATE) 0716 to the E-BUS. 0716 is activated by the state machine (0703) when it is E-BUS MASTER. There is the possibility of dynamically reconfiguring and controlling the INPUT CELL via the unit itself (DFP, FPGA, DPGA, etc.) instead of through the PLU. The I-BUS connection to the EB-REG (0702) and the MODE PLUREG (0708) serves this function.

Figure 8: Figure 8 shows the MODE PLUREG of an INPUT or OUTPUT CELL written by the PLU via the PLU bus (0802) or via an I-BUS (0808). The respective bus system is selected by the multiplexer (0809) (control of the multiplexer is not shown because an ordinary decoder logic can be used). The counter settings such as step length, counting direction and enabling of the counter are sent directly (0807) to the counter (0803). The basic address can either be written directly (0805) into the counter via a load (0804) or stored temporarily in an extension (0811) of 0801. Records in 0801 that are relevant for the state machine go to the state machine via a gate (0806) which is opened by the state machine for the INPUT or OUTPUT CELL activated at the time.

Figure 9a: Figure 9a shows a bus interface circuit with a state machine (0901), MASTER GATE (0902) and EB-REG (0903). The INPUT CELLs (0904) transfer data from the E-BUS (0905) to the II-BUS (0906). The OUTPUT CELLs (0907) transfer data from the IO-BUS (0908) to the E-BUS (0905). All units are linked together by the control bus (0909).

Figure 9b: Figure 9b shows a bus interface circuit with state machine (0901), MASTER GATE (0902) and EB-REG

(0903). The INPUT CELLs (0904) transfer data from the E-BUS (0905) to the bidirectional I-BUS (0910). The OUTPUT CELLS (0907) transfer data from the bidirectional I-BUS (0910) to the E-BUS (0905). All units are linked together over the control bus (0909). Interface circuits which use both possibilities (Figures 9a and 9b) in a hybrid design are also conceivable.

Figure 10a: Figure 10a shows the interconnections of two units (DFPs, FPGAs, DPGAs, etc.) (1001) interconnected via the E-BUS (1002).

Figure 10b: Figure 10b shows the interconnection of a number of units (DFPs, FPGAs, DPGAs, etc.) (1001) via the E-BUS (1002).

Figure 10c: Figure 10c shows the interconnection of a number of units (DFPs, FPGAs, DPGAs, etc.) (1001) via the E-BUS (1002). The interconnection can be expanded to a matrix. One unit (1001) may also manage multiple bus systems (1002).

Figure 10d: Figure 10d shows the interconnection of a unit (DFP, FPGA, DPGA, etc.) (1001) to a memory unit or a memory bank (1003) via the E-BUS (1002).

Figure 10e: Figure 10e shows the interconnection of a unit (DFP, FPGA, DPGA, etc.) (1001) to a peripheral device or a peripheral group (1004) via the E-BUS (1002).

Figure 10f: Figure 10f shows the interconnection of a unit (DFP, FPGA, DPGA, etc.) (1001) to a memory unit or a memory bank (1003) and to a peripheral device or a peripheral group (1004) via the E-BUS (1002).

Figure 10g: Figure 10g shows the interconnection of a

unit (DFP, FPGA, DPGA, etc.) (1001) to a memory unit or a memory bank (1003) and to a peripheral device or a peripheral group (1004) plus another unit (DFP, FPGA, DPGA, etc.) (1001) via the E-BUS (1002).

Figure 11: Figure 11 shows the architecture of the EB-REG. The bus systems E-BUS (1103), the PLU bus (1104) over which the PLU has access to the EB-REG, and the local internal bus between the INPUT/OUTPUT CELLs, the state machine and the EB-REG (1105, see 0407, 0417, 0517) and possibly an I-BUS (1114) are connected to a multiplexer (1106). The multiplexer (1106) selects either one of the buses or the feedback to the register (1108) and switches the data through to the input of the register (1108). The MASTER bit is sent separately over the multiplexer (1107) to the register (1108). The multiplexer is controlled by the RESET signal (1101) (resetting or initializing the unit). If a RESET signal is applied, the multiplexer (1107) switches the signal of an external chip connection (1102) through to the input of the register (1108); otherwise the output of the multiplexer (1106) is switched through to the input of the register (1108). Thus MASTER may be preallocated. The register (1108) is clocked by the system clock (1112). The contents of the register (1108) are switched via a gate (1109, 1110, 1111, 1113) to the respective bus system (1103, 1104, 1105, 1114) having read access at that time. The control of the gates (1109, 1110, 1111, 1113) and of the multiplexer (1106) is not shown because an ordinary decoder logic may be used.

Embodiments: Figure 12 shows an example using the standard bus system RAMBUS (1203). A unit (DFP, FPGA, DPGA, etc.) (1201) is connected to other units (memories, peripherals, other DFPs, FPGAs, DPGAs, etc.) (1202) over the bus system (1203). Independently of the bus system (1203), the unit (1201) may have additional connecting

lines (1204), e.g., as is customary in the related art, for connecting any desired circuits.

Definition of terms: The following is a definition of terms used above.

ADR-GATE: Gate which switches addresses to the E-BUS if the unit is in E-BUS MASTER mode.

DFP: Data flow processor according to German Patent DE 44 16 881.

DPGA: Dynamically programmable gate array. Related art.

D flip-flop: Storage element which stores a signal at the rising edge of a clock pulse.

EB-REG: Register that stores the status signals between I-BUS and E-BUS.

E-BUS: External bus system outside a unit.

E-BUS MASTER: Unit that controls the E-BUS. Active.

E-BUS SLAVE: Unit controlled by the E-BUS MASTER. Passive.

E-GATE: Gate which is controlled by the internal state machine of the unit or by the E-BUS MASTER and switches data to the E-BUS.

E-GATE-REG: Register into which data transmitted to the E-BUS over the E-GATE is entered.

E-READ: Flag in the EB-REG indicating that the OUTPUT CELLS have been transferred completely to the E-BUS.

E-WRITE: Flag in the EB-REG indicating that the E-BUS has been transferred completely to the INPUT CELLS.

Flag: Status bit in a register, indicating a state.

FPGA: Field programmable gate array. Related art.

Handshake: Signal protocol where a signal A indicates a state and another signal B confirms that it has accepted signal A and responded to it.

INPUT CELL: Unit transmitting data from the E-BUS to an I-BUS.

I-BUSn *(also I-BUS)*: Internal bus system of a unit, which may also consist of bundles of individual lines, where n indicates the number of the bus.

II-BUSn *(also II-BUS)*: Internal bus system of a unit, which may also consist of bundles of individual lines, with n denoting the number of the bus. The bus is driven by an INPUT CELL and goes to logic inputs.

IO-BUSn *(also IO-BUS)*: Internal bus system of a unit, which may also consist of bundles of individual lines, with n denoting the number of the bus. The bus is driven by logic outputs and goes to an OUTPUT CELL. n indicates the number of the bus.

I-GATE: Gate that switches data to the I-BUS.

I-GATE-REG: Register which is controlled by the internal state machine or by E-BUS MASTER and into which data transmitted over the I-GATE to the I-BUS is entered.

I-READ: Flag in the EB-REG indicating that the INPUT CELLs have been completely transferred to the I-BUS.

I-WRITE: Flag in the EB-REG indicating that the I-BUS has been completely transferred to the OUTPUT CELLs.

Edge cell: Cell at the edge of a cell array, often with direct contact with the terminals of a unit.

Configuring: Setting the function and interconnecting a logic unit, a (FPGA) cell (logic cell) or a PAE (see reconfiguring).

Primary logic unit (PLU): Unit for configuring and reconfiguring a PAE or logic cell. Embodied by a microcontroller specifically designed for this purpose.

Latch: Storage element which usually relays a signal transparently during the H level and stores it during the L level. Latches where the function of levels is exactly the opposite are sometimes used in PAEs. An inverter is then connected before the clock pulse of a conventional latch.

Logic cells: Configurable cells used in DFPs, FPGAs, DPGAs, fulfilling simple logical or arithmetic functions, depending on configuration.

MASTER: Flag in EB-REG showing that the E-BUS unit is a MASTER.

MODE PLUREG: Register in which the primary logic unit sets the configuration of an INPUT/OUTPUT CELL.

OUTPUT CELL: Unit that transmits data from an I-BUS to the E-BUS.

PAE: Processing array element: EALU with O-REG, R-REG, R20-MUX, F-PLUREG, M-PLUREG, BM UNIT, SM UNIT, sync UNIT, state-back UNIT and power UNIT.

PLU: Unit for configuring and reconfiguring a PAE or a logic cell. Embodied by a microcontroller specifically designed for this purpose.

REQ-MASTER: Flag in the EB-REG indicating that the unit would like to become E-BUS MASTER.

RS flip-flop: Reset/set flip-flop. Storage element which can be switched by two signals.

SET-REG: Register indicating that data has been written in an I-GATE-REG or E-GATE-REG but not yet read.

STATE-GATE: Gate switching the output of the SET-REG to the E-BUS.

Gate: Switch that relays or blocks a signal. Simple comparison: relay.

Reconfiguring: New configuration of any number of PAEs or logic cells while any remaining number of PAEs or logic cells continue their own function (see configuring).

State machine: Logic which can assume miscellaneous states. The transitions between states depend on various input parameters. These machines are used to control complex functions and belong to the related art.

Naming conventions:

Unit: -UNIT
Operating mode: -MODE
Multiplexer: -MUX
Negated signal: not-
Register for PLU visible: -PLUREG
Internal register: -REG
Shift registers: -sft

24

Function convention:

Shift registers: sft

AND function: &

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

T2SOX

OR function: #

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

NOT function: !

A	Q
0	1
1	0

GATE function: G

EN	D	Q
0	0	-
0	1	-

25

EN	D	Q
1	0	0
1	1	1

5